

LA-UR-19-29979

Approved for public release; distribution is unlimited.

Title: Eulerian Applications Project - xRage High Explosives Package

Author(s): Grove, John W.

Intended for: Report

Issued: 2019-10-02

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.



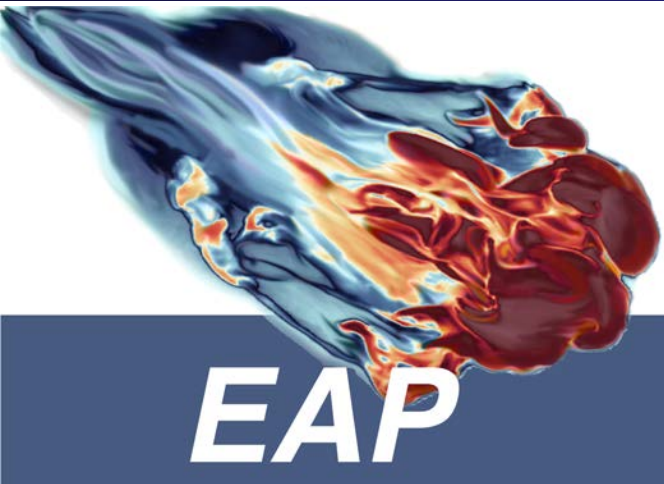


• **Los Alamos**
NATIONAL LABORATORY
— EST. 1943 —

Delivering science and technology
to protect our nation
and promote world stability

Eulerian Applications Project - xRage

High Explosives Package



John W. Grove

CCS-2

September 30, 2019



Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA

LA-UR-19-XXXXX

High Explosive Modeling in xRage

High Explosive Models for Hydro Codes

- **I unashamedly lifted most of the following material from the xRage user manual.**
 - Thanks to Ralph Menikoff who wrote most the HE section of the xRage user manual.
- At the most elementary level, the high explosive model in xRage (or many other hydro codes) computes the amount of material that is changes from a high explosive reactant to a high explosive product during a time step and accounts for the energy added to the flow by this reaction.
- The actual chemical reactions that occur during this conversion are not modeled. These processes are very complicated and involve levels of spatial and temporal resolution that are generally not practical for an application code. They would also require a state variable representation that includes all of the chemical species that are produced by the reaction and account for multiple stage chemical reactions.
- Instead the explosive models we generally use are phenomenological and try to represent the most important physics that drive the explosive wave in a very simplified way. These models generally have a number of tunable parameters that must be calibrated to experiment.
 - One could say a model is a success if this calibration applies to a “wide” variety of modeling situations.
 - There is not much point in a model that has to be recalibrated to every possible situation where it would be used.

High Explosive References

- There is an extensive literature on high explosive modeling. Here are a few good books with which to get started.
- Fickett, W. and W. Davis, *Detonation*. 1979, Berkeley, CA: Univ. of California Press.
- Mader, C., *Numerical Modeling of Detonations*. 1979, Berkeley: University of California Press.
- Mader, C.L., *Numerical Modeling of Explosives and Propellants*. second ed. 1998: CRC Press.
- The above books give the background for many of the HE models used at LANL.
- Courant, R. and K. Friedrichs, *Supersonic Flow and Shock Waves*. 1967, New York: Springer-Verlag. Chapter III section E.
 - This book has a short but nice description of some of the basic mathematical components of detonations and deflagrations.
- A more extensive reference list can be found at the end of the high explosive chapter of the xRage user manual.

Getting Started for HE in xRage

- There are three basic components to the xRage input specifications that are used to add a high explosive component to a simulation
- Equation of State: EOS models for the explosive reactants and products need to be added to the tabular EOS input file.
- Specifications for the desired HE model: these include:
 - The HE model to be used
 - Parameters used by the model. These will be model dependent and may specify things like detonator location and wave speeds.
- Regions containing the high explosive reactants need to be added to the input deck.
 - Nothing will burn if there is nothing to burn.
- Once these input modifications are done you run xRage in the usual way.
- Depending on the HE model being used additional variables and diagnostics will be available in either the restart dump file or the xRage output file.

“Simple” Example: Programmed Burn for PBX-9501

Equation of State Input

- TEOS input
- Sesame models for the PBX-9501 reactants and products. I have two files s950104 for 9501 reactants and sB211 for products.

```
$nses = ($nses+1)
sesfiles($nses) = "s950104" ! 9501 reactants
$nses = ($nses+1)
sesfiles($nses) = "s8211" ! 9501 products
```
- I add them to my teos.in. These would be in addition to any other Sesame files required for the simulation. Here I am using the Parser to keep track of the files added to the sesfiles list.
- Add the materials to the teos.in file
- As before \$mat is a parser variable keeping track of the materials being entered.

```
$mat = ($mat+1)
matident($mat) = '9501reactants'
matid($mat) = 950104 ! 9501 solid
eostype($mat) = 0
$mat = ($mat+1)
matident($mat) = '9501products'
matid($mat) = 8211 ! 9501 products table with energy offset
eostype($mat) = 0
```

“Simple” Example: Programmed Burn for PBX-9501

Input File Modifications

Specify the explosive materials.

! 9501 reactants and products

\$he = (\$he+1)

\$mat = (\$mat + 1)

\$mat9501reactants = \$mat

\$matrl_id_9501reactants = 950104 ! Booster HE 9501 solid

\$Rho0_9501reactants = 1.836 ! g/cc

matident(\$mat9501reactants) = "9501_reactants"

matdef(1,\$mat9501reactants) = \$matrl_id_9501reactants

pmin_mat(\$mat9501reactants) = 0.0

\$mat = (\$mat + 1)

\$mat9501products = \$mat

\$matrl_id_9501products = 8211 ! Booster HE 9501 products

\$Rho0_9501products = 1.836 ! g/cc

matident(\$mat9501products) = "9501_products"

matdef(1,\$mat9501products) = \$matrl_id_9501products

pmin_mat(\$mat9501products) = 0.0

“Simple” Example: Programmed Burn for PBX-9501

Input File Modifications

Specify the explosive model

```
! CJ state 9501
$gamma_9501reactants = 2.68          ! CJ adiabatic index
$Dcj_9501reactants = (8.8*$km_s)      ! CJ detonation speed
$Pcj_9501reactants = ($Rho0_9501reactants*$Dcj_9501reactants**2/($gamma_9501reactants+1)) ! CJ pressure
$Edet = 0                            ! energy offset: products - reactants
! HE rate
he_model($he) = 3                    ! Programmed Burn
he_zone_size($he) = 0.1 !! mm
he_adapt_pres($he) = 1.0e10 ! cutoff pressure for refinement 10 kilobars
! ----- Programmed burn parameters
$xdet = (9.0 * $cm)
$ydet = (5.5 * $cm)
he_detvel($he) = $Dcj_9501reactants ! detonation speed
he_pcj($he) = $Pcj_9501reactants ! CJ pressure
he_det_number($he) = 1              ! number of detonators
he_tau_prz($he) = 1.0              ! pseudo-reaction time constant (dimensionless)
he_det_points(1,$he) = $xdet ! There is a single detonation point at (the point (9, 5.5)
he_det_points(2,$he) = $ydet
he_unreacted($he) = $mat9501reactants ! reactants
he_reacted($he) = $mat9501products ! products
he_number = $he
```

“Simple” Example: Programmed Burn for PBX-9501

Input File Modifications

Specify the explosive region. A 1 cm diameter circle centered at \$xdet and \$ydet (as input on the previous slide)

```
! 9501
```

```
$reg = ($reg + 1)
```

```
matreg($reg) = $mat9501reactants
```

```
typreg($reg) = 1
```

```
diareg(1,$reg) = 1.0
```

```
xreg(1,$reg) = $xdet
```

```
yreg(1,$reg) = $ydet
```

```
prsreg($reg) = 1e6 ! ! bar
```

```
tevreg($reg) = (296.15/11604.506) ! 296.15 Kelvin in units eV/k
```

HE Library Fundamentals

- The high explosive library computes the transformation of one material (HE reactants) into another (HE products) by updating the mass fractions of the reactants and products via the equation

$$\frac{Dw}{Dt} = \frac{\partial w}{\partial t} + \vec{u} \cdot \nabla w = -Rate(w, p, T)$$

- Here w is the mass fraction of reactants relative to the total mass of reactants and products $w = \frac{mass_{reactants}}{mass_{reactants} + mass_{products}}$.
- We define the reaction progress variable $\lambda = 1 - w$ as the mass fraction of the products
 - Note: while the typical case is that a cell will contain only reactants and products for a single material this need not be true in general as other materials can be present in the mixed cell as well. Thus w and λ need not equal the material mass fractions of the reactants and products in a cell.
- The key feature of a high explosive reaction is that the burn front is self-sustaining and will be driven by the energy released by the chemical reactions that convert reactants into products.

HE Energy Release

- The chemical energy inside the reactants is converted into heat by the reaction. This heat release can be modeled in two ways.
 1. The equation of state of the products incorporates this additional energy in its specific internal energy.
 2. The energy release can be represented directly via the entropy equation for the materials

$$T \frac{DS}{Dt} = \frac{De}{Dt} + P \frac{DV}{Dt} = Q \times \text{Rate}(w, p, T)$$

- Here Q is the energy per unit mass released by the chemical reaction
 - Generally Q is a constant parameter.
- It is up to the user to determine which of the two methods for energy release by the reaction is to be used. Be careful not to over count by trying to use both.
 - In practice this requires knowing whether the products EOS already incorporates this energy release. **Caveat Emptor!**

General HE Parameters

The following describe generic parameters used by all HE models in xRage.

Input Parameter	Explanation	Default
he_number	Number of HE models	-1
he_reacted(he)	matid for products	none
he_unreacted(he)	matid for reactants	none
he_model(he)	model type	none
he_rhoz(he)	reference density	0
he_specific_energy(he)	specific energy release on burn (Q)	0
he_energy(he)	energy release per volume (only used if above is 0)	0
he_detvel(he)	CJ detonation velocity	-1 (detvel= D_{CJ} from EOS)
he_dtpct(he)	CFL for HE time step	0.4
he_pmin(he)	Cutoff pressure for reaction	0
he_pscale(he)	Pressure units scale	Mbar = 10^{12} μ bar
he_tscale(he)	Time units scale	μ sec = 10^{-6} sec
he_w_cutoff(he)	cutoff for burning remainder of HE	0.001
he_dw(he)	limit change in change in burn fraction per time step	1.0

HE Mesh control

- The following parameters direct the AMR to refine the mesh at burn fronts
- Of these `he_zone_size(he)` is the most commonly used parameter
- Models such as SURF/SURFplus work best when the reaction zone has sufficient mesh resolution. These parameter allow this control.

Parameter	Description	Default Value
<code>he_adapt_pres(he)</code>	cutoff pressure for HE refinement (μbar)	0
<code>he_size</code>	default for <code>he_zone_size(he)</code> (deprecated)	
<code>he_zone_size(he)</code>	refine active HE cells down to <code>he_zone_size(he)</code> [cm]. Active means $w > 0$ and $\text{pres} > \text{he_adapt_pres(he)}$	0 (skip refinement)
<code>he_refine_dw(he)</code>	refine only until $\text{Rate(he)} \frac{\Delta x}{D_{CJ}} < \text{he_refine_dw(he)}$	0

Ignition and Growth Model: $he_model(he) = 1$

- $\frac{Dw}{Dt} = \frac{\partial w}{\partial t} + \vec{u} \cdot \nabla w = -Rate(w, p, T)$
- $Rate = \frac{(Rate_I + Rate_G + Rate_B)}{t_{scale}}$
- $Rate_I(w, \rho(p, T)) = \begin{cases} I w^b \left(\frac{\rho}{\rho_0} - 1 - a \right)^x & (1 - w) < F_{igmax} \text{ and } \rho \rho_0 > 1 + a0 \\ 0 & \text{otherwise} \end{cases}$
- $Rate_G(w, p) = \begin{cases} G_1 w^c (1 - w)^d \left(\frac{p}{P_{scale}} \right)^y & (1 - w) \leq F_{G1max} \\ 0 & \text{otherwise} \end{cases}$
- $Rate_B(w, p) = \begin{cases} G_2 w^e (1 - w)^g \left(\frac{p}{P_{scale}} \right)^z & F_{G2min} \leq (1 - w) \\ 0 & \text{otherwise} \end{cases}$
- Parameters: $I, b, a, x, F_{igmax}, G_1, c, d, y, F_{G1max}, G_2, e, g, z, F_{G2min}$
 - These parameters are entered as **he_constants(j, he)** in the order above. The index *he* denotes the HE model $1 \leq he \leq he_number$
- $\rho_0 = \mathbf{he_rhoz}(he)$, $P_{scale} = \mathbf{he_pscale}(he)$, $t_{scale} = \mathbf{he_tscale}(he)$
- Reference: E. L. Lee and C. M. Tarver. Phenomenological model of shock initiation in heterogeneous explosives. Phys. Fluids, 23:2362–2372, 1980.

Forest Fire Model: $\text{he_model}(\text{he}) = 2$

$$\text{Rate}(w, p) = \frac{w}{t_{scale}} \exp \left[\sum_{n=1}^N a_n \hat{p}^{N-n} \right], \hat{p} = \frac{\min(p, P_{CJ})}{P_{scale}}$$

- Input:
 - $N = \text{he_num_coef}(i)$, $n_{lo} = 0$, $a_n = \text{he_constants}(n, i)$
 - $P_{CJ} = \text{he_pcj}(i)$ (default 0)
- If the input parameter **he_rate_size**(he) (default 0) is positive and the parameter **he_rate_power**(he) (default 1) is also positive, and the cell size is less than **re_rate_size**(he) the rate function above is scaled by the factor

$$\left[\frac{\text{he_rate_size}(\text{he})}{\text{cell_size}} \right]^{\min(1, \text{he_rate_power}(\text{he}))}$$

- The differential equation for w is integrated over the cycle using an end point quadrature with $2 \cdot \text{nsteps}$ pressure intervals between the pressure at the current and previous cycle. The value for **nsteps** is set by the input directive **he_nsteps**(he) (default 10).

Forest Fire Extensions

- Two extensions of the Forest Fire model are implemented in xRage.
 - Multiple-Shock Forest Fire aimed at modeling shock desensitization
 - Forest Fire with build up models the transient increase in the CJ pressure using an intermediate reaction.
- Technical details are these extensions can be found in the following references
 - *Review of the Forest fire model for high explosives*, R. Menikoff and M.S. Shaw, *Combustion Theory and Modelling*, Vol. 12, No. 3, June 2008, 569–604
 - *Notes on the Nobel code: Forest fire model, the missing manual*, R. Menikoff, LA-UR-07-6082
 - *Notes on the Nobel code: Implementation of the Forest fire model*, R. Menikoff, LA-UR-07-0559
- The following two slides will summary the user input directives to utilize these two extensions.

Multiple Shock Forest Fire

- Aimed at modeling shock desensitization
- Adds an Arrhenius rate for the chemical reaction (see following)
- Input parameters
 - `he_pcrush(he)` – shock threshold for the weak shock that desensitizes the HE, [*μbar*] default = 0
- There is no standard for how to set these parameters and users will need to use experts to help determine their possible values

Forest Fire with Build-up

- Models the transient increase in CJ pressure with distance to run
- Uses a two reaction model

reactants \rightarrow *intermediate products* \rightarrow *final products*

- Three mass fractions:

$$w_r = \frac{m_r}{m_{he}}, w_{ip} = \frac{m_{ip}}{m_{he}}, w_{fp} = \frac{m_{fp}}{m_{he}}, m_{he} = m_r + m_{ip} + m_{fp}$$

- Three rate equations:

$$\frac{Dw_r}{Dt} = -Rate_1, \frac{Dw_{ip}}{Dt} = Rate_1 - Rate_2, \frac{Dw_{fp}}{Dt} = Rate_2$$

- $Rate_1$ is the standard Forest fire rate

Forest Fire with Build-up

- Three available models for $Rate_2$
- Defined in terms of the mass fraction for the intermediate products

$$w_2 = \frac{m_{ip}}{m_{ip} + m_{fp}}$$

- Gittings impulse model:

$$\frac{DI}{Dt} = \max(0, P - P_{build}), \frac{Dw_2}{Dt} = -\frac{\max(0, P - P_{build})}{I_0 + I(t)} w_2$$

- Simple time constant:

$$\frac{Dw_2}{Dt} = -\tau^{-1} w_2^2, P > P_{build}$$

- Pressure dependent time constant:

$$\frac{Dw_2}{Dt} = -\frac{P}{P_{build}} \tau^{-1} w_2^2, P > P_{build}$$

Forest Fire with Build-up Parameters

- $\text{he_build_mtrl}(\text{he})$ – material index of the final HE products
- $\text{he_build_flag}(\text{he})$ – build up submodel option (default 0 do not use the build-up model)
- $\text{he_build_fmin}(\text{he})$ – minimum mass fraction of final products (default 0)
- $\text{he_build_cutoff}(\text{he})$ – cutoff for completing the second reaction (default 0)
- $\text{he_build_active}(\text{he})$ - logical switch if false the second reaction is cutoff when all reactants are burned (default true)
- $\text{he_build_pres}(\text{he})$ – the value for P_{build} [μbar]
- Gittings impulse model: $\text{he_build_flag}(\text{he}) = 0$
 - $\text{he_build_coef}(\text{he})$: [$\mu\text{bar sec}$] (default 0)
- Simple time constant: $\text{he_build_flag}(\text{he}) = 1$
- Pressure dependent time constant: $\text{he_build_flag}(\text{he}) = 2$
 - $\text{he_build_coef}(\text{he})$: the value for τ^{-1} [sec^{-1}], (default 0)

Programmed Burn: `he_model(he) = 3`

- Simple line-of-sight model for a burn front that propagates with constant velocity

$$t_{\text{burn}}(\vec{x}) = \min_i \left\{ \frac{|\vec{x} - \vec{x}_i|}{D_{CJ}} + t_i \right\}$$

- `he_det_number(he)` – number of detonators
- `he_det_points(3(j - 1) + 1, he)` – detonator locations, x_j, y_j, z_j
- `he_det_time(j, he)` – detonator time [sec] (default 0)
- `he_tau_prz(he)` – pseudo-reaction time constant, τ_{PRZ} (default 1)
– gives the reaction a width of at least a few cells

$$Rate(\vec{x}, t) = \begin{cases} w(\vec{x})/\tau & , t > t_{\text{burn}}(\vec{x}) \\ 0 & , \text{otherwise} \end{cases}, \tau = \max\left(\tau_{\min} t_{\text{scale}}, \frac{\tau_{PRZ} \Delta x}{D_{CJ}}\right)$$

- `he_tau_min(he)` – τ_{\min} (default 0)

Arrhenius rate model: `he_model(he) = 4`

$$\text{Rate}(w, T) = \frac{w^n}{t_{scale}} \min \left(Z \exp \left[-\frac{T_a}{\min(T, T_{max})} \right], k_{max} \right), T > T_{min}$$

- For $T < T_{min}$ the Arrhenius rate is set to zero
- `he_estar(he)` – activation energy [cal/mole] default = 0, $T_a = e^*/R$ where R is the universal gas constant.
- `he_AR_n(he)` – the exponent n above, (default 1)
- `he_z(he)` – the factor Z above, default = 0
- `he_atevmin(he)` – the factor T_{min} above, [eV/k] default = 0
- `he_AR-Ta(he)` - activation temperature in Kelvin (default 0)
- `he_AR-Tmax(he)` – defines a ceiling (T_{max}) for the Arrhenius rate term above (default 0)
- `he_AR_kmax(he)` – alternate way of specifying a rate ceiling (default ∞)
- Generally the user will only set one of `he_estar(he)` and `he_AR-Ta(he)` and also for `he_AR-Tmax(he)` and `he_AR_kmax(he)`

SURF: `he_model(he) = 6`

- SURF is an acronym for Scaled-Unified-Reactive-Front
 - Invented by R. Menikoff and S. Shaw
- SURF and SURFplus are members of a broad category of reactive burn models that use the ignition and growth concept
- In xRage the burn front is detected using the Hugoniot function

$$H = e - e_0 - \left[\frac{P + P_0}{2} \right] (V_0 - V)$$

- Shock detection is initiated when H becomes positive
 - The lead shock is detected when H changes sign or pressure or density decrease
 - When a shock is detected in a cell the value P_{sh} is set, which is subsequently advected by the flow $\frac{DP_{sh}}{Dt} = 0$.
- For more details on the shock detection algorithm see
 - R. Menikoff. Shock detector for SURF model. Technical Report LA-UR-16-20116, Los Alamos National Lab., 2016.
URL <http://www.osti.gov/scitech/servlets/purl/1234496>.

SURF: $\text{he_model}(\text{he}) = 6$

- SURF is an acronym for Scaled-Unified-Reactive-Front
 - Invented by R. Menikoff and S. Shaw
- SURF and SURFplus are members of a broad category of reactive burn models that use the ignition and growth concept
- The burn front is detected using the Hugoniot function

$$H = e - e_0 - \left[\frac{P + P_0}{2} \right] (V_0 - V)$$

- The zero subscript corresponds to the state of the undisturbed explosive
 - The quantity H is identically zero in this region
- For more details on the shock detection algorithm see
- R. Menikoff. Shock detector for SURF model. Technical Report LA-UR-16-20116, Los Alamos National Lab., 2016.
URL <http://www.osti.gov/scitech/servlets/purl/1234496>.

SURF: Burn propagation

- Once the shock detector is activated burn is propagated by the equation

$$\lambda = g(s)$$

where λ is the reaction progress variable (mass fraction of the reaction products), $g(s)$ is a reaction-scale function and s is a dimensionless reaction-scale variable

$$\frac{Ds}{Dt} = f(p_{sh}) \begin{cases} 0, & p \leq 0 \\ \left[\frac{p}{p_{sh}}\right]^n, & 0 < p < p_{sh} \\ \left[\frac{p}{p_{sh}}\right]^{n_{hi}}, & p_{sh} \leq p \end{cases}$$

– The function $f(p_{sh})$ is the reaction strength function.

- The reaction-scale function is given by

$$g(s) = \frac{1}{g_n(s_1)} \begin{cases} g_0(s), & s \leq s_1 \\ g_0(s_1) + g'_0(s_1)(s - s_1) + \frac{1}{2} g''_0(s_1)(s - s_1)^2, & s_1 < s < s_{cutoff} \\ g_n(s_1), & s > s_{cutoff} \end{cases}$$

$$g_0(s) = 1 - e^{-s^2}$$

$$g_n(s_1) = g_0(s_1) - \frac{1}{2} \frac{[g'_0(s_1)]^2}{g''_0(s_1)}, s_{cutoff} = s_1 - g'_0(s_1)/g''_0(s_1)$$

- These formulas introduce three dimensionless parameters for the reaction scale evolution equation and the reaction-scale function: n , n_{hi} , and s_1

SURF Fitting Form 3

- Fitting forms 0-2 are deprecated. See the xRage user guide
- Form 3 improves fit to Pop plots showing nonlinearity

$$f(p_{sh}) = \frac{F(P_{sh})}{t_{scale}}, P_{sh} = \frac{p_{sh}}{p_{scale}}, P_0 = \frac{p_0}{p_{scale}}, P_{low} = \frac{p_{low}}{p_{scale}}, P_1 = \frac{p_1}{p_{scale}}$$

$$F(P_{sh}) = C \begin{cases} P_{low}^{f_n} \left[\frac{P_{sh} - P_0}{P_{low} - P_0} \right]^{[1 - P_0/P_{low}]f_n} & P_0 < P_{sh} \leq P_{low} \\ P_{sh}^{f_n} & P_{low} < P_{sh} \leq P_1 \\ P_1^{f_n} \left[1 + df_1 \left(1 - e^{-\frac{f_n}{df_1} \frac{P_{sh} - P_1}{P_1}} \right) \right] & P_1 < P_{sh} \end{cases}$$

- Parameters: $C, P_0, P_{low}, P_1, f_n, df_1$
- Input:
 - $C = \text{he_surf_C}(\text{he})$
 - $P_0 = \text{he_surf_P0}(\text{he}), P_{low} = \text{he_surf_Plow}(\text{he}), P_1 = \text{he_surf_P1}(\text{he})$
 - $f_n = \text{he_surf_fn}(\text{he}), df_1 = \text{he_surf_df1}(\text{he})$

SURF Fitting Form 4

- Modify approach to asymptotic rate at high pressure

$$f(p_{sh}) = \frac{F(p_{sh})}{t_{scale}}, P_{sh} = \frac{p_{sh}}{p_{scale}}, P_0 = \frac{p_0}{p_{scale}}, P_{low} = \frac{p_{low}}{p_{scale}}, P_1 = \frac{p_1}{p_{scale}}, P_{high} = \frac{p_{high}}{p_{scale}}$$

$$F(P_{sh}) = C \begin{cases} 0 & P_{sh} \leq P_0 \\ P_{low}^{f_n} \left[\frac{P_{sh} - P_0}{P_{low} - P_0} \right]^{1 - \frac{P_0}{P_{low}}} f_n & P_0 < P_{sh} \leq P_{low} \\ P_{sh}^{f_n} & P_{low} < P_{sh} \leq P_1 \\ P_1^{f_n} \exp \left[f_n \log\left(\frac{P_{sh}}{P_1}\right) \cdot \left(1 - \left[2 \log\left(\frac{P_{high}}{P_1}\right)\right]^{-1} \log\left(\frac{P_{sh}}{P_1}\right)\right) \right] & P_1 < P_{sh} \leq P_{high} \\ P_1^{f_n} \exp \left[\frac{1}{2} f_n \log\left(\frac{P_{high}}{P_1}\right) \right] & P_{high} < P_{sh} \end{cases}$$

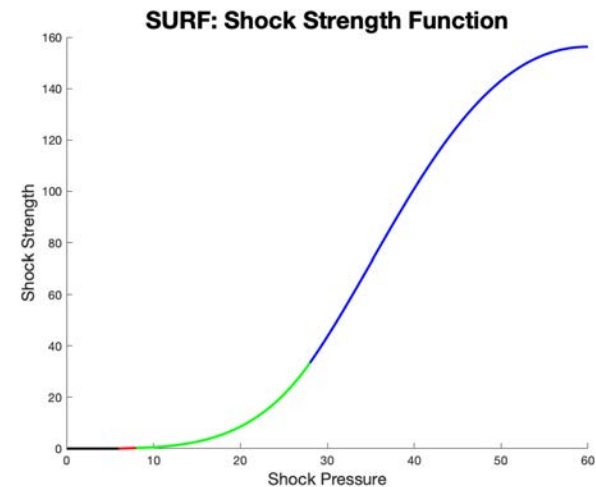
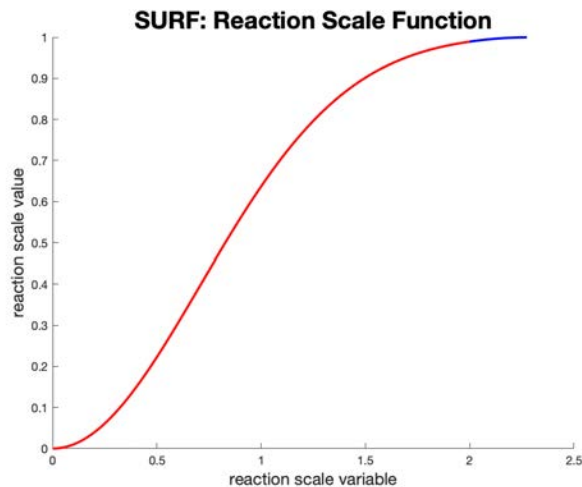
- Parameters: $C, P_0, P_{low}, P_1, P_{high}, f_n, df_1$

- Input:

- $C = \text{he_surf_C}(\text{he})$
- $P_0 = \text{he_surf_P0}(\text{he}), P_{low} = \text{he_surf_Plow}(\text{he}), P_1 = \text{he_surf_P1}(\text{he}), P_{high} = \text{he_surf_Phigh}(\text{he})$
- $f_n = \text{he_surf_fn}(\text{he}), df_1 = \text{he_surf_df1}(\text{he})$

SURF: Model Curves

- The figures below show the graphs of the reaction-scale function and shock-strength function used by the SURF model. The curves used the parameters from the table on the previous slide.
- The sections of the curves are color coded by region
 - Reaction Scale Function: red $s \leq s_1$, blue $s_1 < s \leq s_{cutoff}$
 - Shock-Strength Function: black $\frac{p_{sh}}{p_{scale}} < P_0$, red $P_0 \leq \frac{p_{sh}}{p_{scale}} < P_{low}$, green $P_{low} \leq \frac{p_{sh}}{p_{scale}} < P_1$, blue $P_1 \leq \frac{p_{sh}}{p_{scale}} < P_{high}$



SURF: Shock Detector Knobs

- Several controls are provided to tune the shock detector
- Generally the defaults are adequate for users
- `he_shock_dx(he) [he_zone_size(he)]` sets the mesh refinement value for the shock initiation zone
- `he_shock_Pmin(he) [0.1 GPa]` – threshold pressure to start the shock detection algorithm
- `he_shock_Pweak(he) [0.5 GPa]` – weak shocks have behind pressures below this tolerance and are not detected as burn initiation
- `he_shock_Hzero_tol(he) [10-6]` –
if $\bar{H} < -\text{he_shock_Hzero_tol(he)})(e_1 - e_0)$ reset H to zero
- `he_shock_Hweak_tol(he) [0.01]` –
if $\bar{H} < -\text{he_shock_Hweak_tol(he)})(e_1 - e_0)$ detect a compressive wave or thermal hot spot
- `he_shock_nprofile(he) [10]` – maximum number of zones in a shock profile, shocks that are spread over more zones are not detected
- `he_shock_advect(he) [true]` – turn off advection for time and pressure of the initial shock
- `he_shock_PopPlot(he) [false]` – print diagnostics for printing Pop plots (requires 1D and one processor)

SURFplus: $\text{he_model}(\text{he}) = 7$

- SURFplus is an extension of the SURF model that adds a second slower reaction the follows a burn using SURF
 - The fast reaction corresponds to molecular decomposition and leads to intermediate products.
 - The second reaction burns the intermediate products to produce the final products
- The model system is coupled to the SURF model for the fast reactions via a streamwise ODE and is summarized by the reaction progress variable for intermediate and final products (λ) (mass fractions for intermediate and final products) and relative mass fraction of the final products with respect to the intermediate and final products (λ_2).
- The energy deficit compared to the equilibrium mixture is modeled as

$$e_{cc}(\lambda_2) = \left(\left[\frac{1}{1 + (N_{ratio} - 1)\lambda_2} \right]^{1/3} - N_{ratio}^{-1/3} \right) Q$$

- Equation of state evaluations use the energy offset

$$p(V, e, \lambda, \lambda_2) = p(V, e - \lambda e_{cc}(\lambda_2), \lambda)$$

Where $p(V, \cdot, \lambda)$ is the pressure-temperature equilibrium solution for a mixture of reactants and final products.

SURFplus: Evolution equations

- The evolution equation for λ_2 is modeled as

$$\frac{Ds_2}{Dt} = \frac{1}{2} \lambda_2 \begin{cases} \frac{h_1 a_2}{h_1 s_2 + (t_1 a_2 - h_1)(h_1^{1/2} - s_2)} & s_2^2 < h_1 \\ a_2/s_2 & h_1 \leq s_2^2 \leq h_2 \\ [2(1 - s_2^2) a_3]^{1/2}/s_2 & h_2 < s_2^2 \end{cases}$$

- $s_2 = \sqrt{\lambda_2}$, $a_2 = \frac{h_2 - h_1}{t_2 - t_1}$, $a_3 = \frac{a_2^2}{2(1 - h_2)}$

- Input

- $\text{he_surfplus_Nratio}(\text{he}) = N_{ratio}$
- $\text{he_surfplus_Q}(\text{he}) = Q$
- $\text{he_surfplus_t1}(\text{he}) = t_1$, $\text{he_surfplus_h1}(\text{he}) = h_1$
- $\text{he_surfplus_t2}(\text{he}) = t_2$, $\text{he_surfplus_h2}(\text{he}) = h_2$

WSD – he_model(he) = 8

- Wescott, Stewart, Davis model: extension of the Ignition and Growth model
 - Single parameter set corresponds to both initiation and propagation
 - With sufficient resolution reproduces the curvature effect
 - can account for shock desensitization
- Recall $w = 1 - \lambda$ where λ is the reaction progress variables

$$\frac{Dw}{Dt} = \begin{cases} -\frac{Rate(w, P, \rho)}{t_{scale}} & \psi < 1 \text{ and } w > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{D\psi}{Dt} = \begin{cases} \frac{k}{t_{scale}} \frac{p}{p_{scale}} & P_{hel} \leq \frac{p}{p_{scale}} \leq P_{max} \text{ and } \lambda < \lambda_{psi} \\ 0 & \text{otherwise} \end{cases}$$

$$Rate(w, p, \rho)$$

$$= S_{ign}(\lambda)r_{ign}(w, \rho) + S_{grw}(\lambda)r_{grw}(w, p) + [1 - S_{grw}(\lambda)]r_{bo}(w, p)$$

WSD –Flow Model Continued

$$S_{ign}(\lambda) = \frac{1}{2} \left[1 - \tanh \left(f_{ign}(\lambda - \lambda_{ign}) \right) \right], S_{grw}(\lambda) = \frac{1}{2} \left[1 - \tanh \left(f_{grw}(\lambda - \lambda_{grw}) \right) \right]$$

$$r_{ign}(w, \rho) = \begin{cases} K_{ign} w^b \left(\frac{\rho}{\rho_0} - 1 - a \right)^x & \frac{\rho}{\rho_0} > 1 + a \\ 0 & \text{otherwise} \end{cases}$$

$$r_{grw}(w, p) = w^c (1 - w)^d [S_{id}(\rho_{sh}) r_{ig}(p) + (1 - S_{id}(\rho_{sh})) r_{dg}(p)]$$

$$\frac{D\rho_{sh}}{Dt} = \begin{cases} \frac{D\rho}{Dt} & \rho > \rho_{sh} \text{ and } 1 - w = \lambda < \lambda_{rho} \\ 0 & \text{otherwise} \end{cases}$$

$$S_{id}(\rho_{sh}) = \frac{1}{2} \left[1 - \tanh \left(f_{id} \left(\frac{\rho_{sh}}{\rho_c} - 1 \right) \right) \right]$$

$$r_{ig}(p) = K_{ig} \left(\frac{p}{p_{scale}} \right)^y, r_{dg}(p) = K_{dg} \left(\frac{p}{p_{scale}} \right)^n, r_{bo}(w, p) = K_b w^e (1 - w)^g \left(\frac{p}{p_{scale}} \right)^z$$

WSD - input

- Ignition Terms:

- $f_{ign} = \text{he_wsd_fign}(\text{he})$, $\lambda_{ign} = \text{he_wsd_lambda_ign}(\text{he})$
- $K_{ign} = \text{he_wsd_Kign}(\text{he})$
- $a = \text{he_wsd_a}(\text{he})$, $b = \text{he_wsd_b}(\text{he})$, $x = \text{he_wsd_x}(\text{he})$

- Growth Terms:

- $f_{grw} = \text{he_wsd_fgrw}(\text{he})$, $\lambda_{grw} = \text{he_wsd_lambda_grw}(\text{he})$
- $c = \text{he_wsd_c}(\text{he})$, $d = \text{he_wsd_d}(\text{he})$, $f_{id} = \text{he_wsd_fid}(\text{he})$
- $\rho_c = \text{he_wsd_rho_c}(\text{he})$, $K_{ig} = \text{he_wsd_Kig}(\text{he})$, $y = \text{he_wsd_y}(\text{he})$
- $K_{dg} = \text{he_wsd_Kdg}(\text{he})$, $n = \text{he_wsd_n}(\text{he})$

- Burn out terms

- $K_b = \text{he_wsd_Kb}(\text{he})$, $e = \text{he_wsd_e}(\text{he})$, $g = \text{he_wsd_g}(\text{he})$, $z = \text{he_wsd_z}(\text{he})$

- Desensitization

- $k = \text{he_wsd_k}(\text{he})$, $P_{hel} = \text{he_wsd_Phel}(\text{he})$, $P_{\max} = \text{he_wsd_Pmax}(\text{he})$
- $\lambda_{psi} = \text{he_wsd_lambda_psi}(\text{he})$

- Lead Shock Density

- $\lambda_{rho} = \text{he_wsd_lambda_rho}(\text{he})$

AWSD – he_model(he) = 9

- The Arrhenius WSD model aims at having the reaction rate account for the initial HE temperature.
- $Rate(w, p, T_{sh}) = F_p(p)[F_1(p, T_{sh}, \lambda) + F_2(T_{sh}, \lambda)]F_\lambda(\lambda)$
- $\frac{dT_{sh}}{Dt} = 0$, if $\lambda < 0.5$ and $\zeta < 1$ then $T_{sh} = \max(T_{sh}, T_*(T, \lambda))$
- $T_*(T, \lambda) = T \left[1 - \alpha_T e^{\lambda \frac{T_c}{T}} \right]$, $\frac{D\zeta}{Dt} = \begin{cases} k_\zeta & \text{if } p > p_\zeta \\ 0 & \text{otherwise} \end{cases}$
- $F_p(p) = \begin{cases} e^{-\left(\frac{p_s}{p}\right)^{np}} & \text{if } p > p_\zeta \\ 0 & \text{otherwise} \end{cases}$
- $F_1(p, T_{sh}, \lambda) = k_1 e^{-\frac{T_1}{T_{sh}}} \left(\lambda + a_1 F_p(p) \right) (1 - \lambda)^{b_1}$
- $F_2(T_{sh}, \lambda) = k_2 e^{-\frac{T_2}{T_{sh}}} (1 - \lambda)^{b_2}$
- $F_\lambda(\lambda) = \begin{cases} f_s + \frac{1}{2} (1 - f_s) \left[1 - \tanh \left(\frac{\lambda - \lambda_c}{\delta_\lambda} \right) \right] & \lambda < 1 \\ 0 & \text{otherwise} \end{cases}$

AWSD - input

- $\alpha_T = \text{he_awsd_alphaT}(\text{he})$ - dimensionless
- $T_c = \text{he_awsd_Tc}(\text{he})$ - units Kelvin
- $k_\zeta = \text{he_awsd_kzeta}(\text{he})$ – units $1/t_{scale}$
- $p_\zeta = \text{he_awsd_pzeta}(\text{he})$ – units p_{scale}
- $n_p = \text{he_awsd_np}(\text{he})$ – dimensionless
- $p_s = \text{he_awsd_ps}(\text{he})$ – units p_{scale}
- $k_1 = \text{he_awsd_k1}(\text{he})$ - units $1/t_{scale}$
- $a_1 = \text{he_awsd_a1}(\text{he})$ – dimensionless
- $b_1 = \text{he_awsd_b1}(\text{he})$ – dimensionless
- $k_2 = \text{he_awsd_k2}(\text{he})$ - units $1/t_{scale}$
- $T_2 = \text{he_awsd_T2}(\text{he})$ - units Kelvin
- $b_2 = \text{he_awsd_b2}(\text{he})$ – dimensionless
- $f_s = \text{he_awsd_fs}(\text{he})$ – dimensionless
- $\lambda_c = \text{he_awsd_lambdaC}(\text{he})$ – dimensionless
- $\delta_\lambda = \text{he_awsd_Dlambda}(\text{he})$ - dimensionless

xRage HE Models

- We have outlined the various high explosive models available in the xRage code
- Generically, each model computes the transformation of one chemical species (reactants) into another (products) and handles the energy release by the reaction (either explicitly or implicitly via the products equations of state)
- For a user, the big question is how to determine appropriate values for the model parameters and which models to use for the reactants and products equation of state
- As Menikoff points out in the xRage user manual the combination of an HE model and reactants/products equation of state is not arbitrary. HE model parameters are calibrated to experiment based the a specific choice of the the HE equations of state. Mixing and matching HE model parameters with HE EOS models and parameters is a possible source of error.
- Users should seek to use values taken from LANL HE databases when available and otherwise to seek expert knowledge is the selection of the proper HE model with the accompanying EOSs.